SPECIFICATION

Electronic Version 1.2.8

Stylesheet Version 1.0

METHOD AND SYSTEM FOR RESOLVING UNIVERSAL RESOURCE LOCATORS (URLs) FROM SCRIPT CODE

Background of Invention

[0001] This invention relates to a method and system for resolving Universal Resource Locators (URLs) from script code located in websites for the purpose of website crawling.

[0002] The World Wide Web available on the Internet provides a variety of specially formatted documents called web pages. The web pages are traditionally formatted in a language called HTML (HyperText Markup Language). Many web pages include links to other web pages which may reside in the same website or in a different website, and allow users to jump from one page to another simply by clicking on the links. The links use Universal Resource Locators (URLs) to jump to other web pages. URLs are the global addresses of web pages and other resources on the World Wide Web.

[0003] As web technology evolves, websites become more and more complex. The tendency in website development is to move from using purely static HTML to using HTML and script code to provide enhanced functionality. As a result, it is now common to use script code to construct web page links, i.e., to create URLs dynamically. Often the process of dynamically constructing URLs involves many variables and some rather complex script code. This makes it very difficult to resolve, i.e., extract and obtain, such URLs, when it comes to website crawling.

[0004] Website crawling or spidering is a process to automatically scan contents of websites by following links and fetching the web pages. Web crawling agents or

"spiders" are software programs for performing the crawling over websites. Typically, existing web crawling agents are used to find specific information of interest in the Web.

[0005] Before the introduction of script code into Web pages, crawling agents could parse HTML code for standard URLs. Since all URLs had to be coded to the HTML specification, this task was relatively easy. However, as sites evolved they increasingly relied upon script code to provide more advanced functionality that standard HTML did not allow for. The format of the URLs in the script code varies widely from implementation to implementation. Unlike static HTML, there is no standard that the script code must follow for encoding URLs. Accordingly, script code presents problems for crawling agents that need to parse URLs. There is no longer a common syntax or format for the URLs and thus they are difficult to find consistently.

[0006] An existing approach to this problem is to use customizable pattern matching algorithms that statically read through the script code on a page or in a script file, and based on pattern matching try to "guess" what in that script code might be a URL. The pattern matching provides some utility but the use of the pattern matching algorithms has two basic problems: 1) the algorithms invariably miss URLs in the script code and 2) the algorithms do not always extract the entire URL correctly.

[0007] It is therefore desirable to provide a new mechanism that can more accurately resolve URLs from script code embedded in web pages.

Summary of Invention

[0008] It is an object of the invention to provide a novel system and method for more accurately resolving Universal Resource Locators (URLs) from script code.

[0009] The present invention examines the script code to obtain URLs from the output in the context of website crawling.

[0010]

In accordance with an aspect of the present invention, a URL resolution system for resolving Universal Resource Locators (URLs) is provided. The URL resolution system comprises a website crawler for crawling a website and for locating script code which is used to dynamically create at least one script URL; and a script URL resolution

component for causing examination of the script code located during the crawling to obtain the script URL.

[0011] In accordance with another aspect of the present invention, a method for resolving Universal Resource Locators (URLs) is provided. The method comprises steps of locating script code which creates at least one script URL while crawling a website; and examining the script code to obtain the script URL from the examination result.

[0012] In accordance with another aspect of the present invention, a computer readable medium storing the instructions and/or statements for use in the execution in a computer of the method for resolving Universal Resource Locators (URLs) is provided.

[0013] In accordance with another aspect of the present invention, there is provided electronic signals for use in the execution in a computer of the method for resolving Universal Resource Locators (URLs).

[0014] In accordance with another aspect of the present invention, there is provided a computer program product for use in the execution in a computer of a method for resolving Universal Resource Locators (URLs). The computer program product comprises a module for locating script code which creates at least one script URL while crawling a website; and a module for examining the script code to obtain the script URL from the examination result.

[0015] Other aspects and features of the present invention will be readily apparent to those skilled in the art from a review of the following detailed description of preferred embodiments in conjunction with the accompanying drawings.

Brief Description of Drawings

[0016] The invention will be further understood from the following description with reference to the drawings in which:

[0017] Figure 1 is a diagram showing an example of websites having script code;

[0018] Figure 2 is a block diagram showing a URL resolution system in accordance with an embodiment of the present invention;

[0019] Figure 3 is a flowchart showing a method for resolving a URL in accordance with

an embodiment of the present invention;

[0020] Figure 4 is a block diagram showing a URL resolution system in accordance with another embodiment of the present invention;

[0021] Figure 5 is a block diagram showing a URL resolution system in accordance with another embodiment of the present invention;

[0022] Figure 6 is a flowchart showing a method for resolving a URL in accordance with another embodiment of the present invention;

[0023] Figure 7 is a flowchart showing a method for resolving a URL in accordance with another embodiment of the present invention;

[0024] Figure 8 is a block diagram showing a URL resolution system in accordance with another embodiment of the present invention;

[0025] Figure 9 is a block diagram showing a URL resolution system in accordance with another embodiment of the present invention;

[0026] Figure 10 is a block diagram showing a URL resolution system in accordance with another embodiment of the present invention; and

[0027] Figure 11 is a block diagram showing a URL resolution system in accordance with another embodiment of the present invention.

Detailed Description

[0028] The present invention is suitably used to check the integrity of links in a website. For example, a website 10 shown in Figure 1 contains web pages or documents 20, some of which have embedded script code 30 which is used to dynamically create URLs. URLs created by the script code are called script URLs hereinafter. Each script URL may designate a local web page located within the same website or a remote web page located in a different website.

[0029] For example, in Figure 1, page 2 of website 1 has script code a which is used to create a script URL identifying page 2 of website 2; page 3 of website 1 has script code b which is used to create a script URL identifying page 5 of website 1, and so on.

More than one set of script code may be embedded in a single web page. A single set of script code may create one or more script URLs. The script code typically has a specific part that is used to create one or more script URLs. The entire script code may form the specific part.

[0030] Script code for dynamically creating script URLs may be JavaScript, JScript or VBScript and others.

[0031] Figure 1 schematically represents that script code a in page 2 of website 1 can be successfully resolved to create link 40 to page 2 of website 2. However, script code c in page 3 of website 1 cannot be successfully resolved because of an error in the script code or other reasons, and accordingly, the link as represented by broken arrow 50 is unresolvable.

[0032] Figure 2 shows a URL resolution system 100 in accordance with an embodiment of the present invention. The URL resolution system 100 comprises a website crawler 120 and a script URL resolution component 140. As shown in Figure 3, the website crawler 120 scans or crawls website 110 (200). When it encounters or locates script code in the website 110 that is used to dynamically create one or more script URLs (202), the script URL resolution component 140 causes examination of the script code to resolve its script URL or URLs (204). From the examination output, the script URLs are obtained (206). The crawling is continued to locate any other script code that is used to dynamically create one or more URLs (208).

[0033] The examination of script code at step 204 may be carried out by explicitly executing the script code. Alternatively, it may be done by examining the script code to obtain the script URLs without explicitly executing the script code. The script URL resolution component 140 may examine the script code or it may use another component to examine the script code, as described below in relation with other embodiments.

[0034] Therefore, the URL resolution system 100 allows automatic resolution of script URLs from embedded script code in websites in the context of website crawling, i.e.,by locating script code while crawling a website or websites. Since the script code is examined to dynamically obtain the script URLs, complete URLs can be accurately

obtained. Unlike the conventional pattern matching which resolve URLs statically, there are minimal possibilities that the URL resolution system 100 will miss script URLs in the website that is being crawled. Thus, the URL resolution system 100 produces accurate results of website crawling.

[0035] The URL resolution system 100 may have a function by which users can set the extent of the crawling, as described below.

[0036] Other embodiments of the present invention are described referring to Figures 4 and 5. A URL resolution system 300 shown in Figure 4 comprises a website crawler 320 and a script URL resolution component 340.

[0037] The website crawler 320 has script code detector 322 and crawling controller 324. OLE_LINK2The crawling controller 324 OLE_LINK2controls crawling carried out by the website crawler 320. The crawling controller 324 controls the website crawler 320 to crawl individual web pages included in website 310 or other websites to locate web pages that use script code to dynamically create script URLs. The crawling controller 324 receives output of the script URL resolution component 340 and uses the output to control the website crawler 320, as further described below.

[0038] To locate web pages that use script code to dynamically create script URLs, the website crawler 320 uses the script code detector 322 to determine if the script code contained in the web page should be executed by determining if it uses a specific part of the script code to dynamically create at least one script URL. The script code detector 322 issues a notification to the script URL resolution component 340 when a web page having such script code is found. The notification includes an identification of the web page.

[0039] The script URL resolution component 340 is activated in response to a notification generated by the script code detector 322 of the website crawler 320. The website crawler 320 crawls all web pages on the original website, but it only passes the web pages containing relevant script code to the script URL resolution component 340.

[0040] The script URL resolution component 340 controls a web page examiner 360. The web page examiner 360 is a component capable of loading the contents of web pages and executing the entire or a specific part of script code in the loaded web pages. The

web page examiner 360 may be a web browser having these functions, or a combination of a web page parser and a script code examiner. The URL resolution system 300 uses an external web page examiner 360. Alternatively, as shown in Figure 5, an internal web page examiner 460 may be provided within the URL resolution system 400.

[0041] The script URL resolution component 340 has a web page loading controller 342 and a script code execution controller 344. The web page loading controller 342 notifies or instructs the web page examiner 360 to load relevant web pages. The script code execution controller 344 instructs the web page examiner 360 to execute specific parts of the script code that will result in dynamically created script URLs. For example, when the script URL resolution component 340 receives a notification from the script code detector 322, the web page loading controller 342 instructs the web page examiner 360 to load the contents of the web page identified in the notification. Then, the script code execution controller 344 executes the script code by interfacing with the web page examiner 360 and using its interface functions to force the execution of the specific parts of the script code in the loaded web pages. The web page examiner 360 captures the script URL(s) resulting from the script code execution and returns these script URLs to the script code execution controller 344. The script code execution controller 344 may instruct the web page examiner 360 to execute the entire script code, rather than only the specific parts thereof.

[0042] The script code execution controller 344 outputs the execution results to the website crawler 320. When the execution of the script code is successful, the execution result includes one or more resolved script URLs. When the execution of the script code is unsuccessful, the execution result includes a failure result.

[0043] The URL resolution systems 300, 400 may also have a presentation unit 480 or use an external presentation unit 380 to present to users the execution results. The presentation unit 380, 480 may be a user interface, a result log file, an email or other output unit or form. The execution results presented to users may include only the failure results or only resolved script URLs or both. Thus, an administrator of the website may attend to the failures.

[0044] Users may also use an input unit (not shown) to initiate or terminate the crawling,

or set parameters of the crawling controller 324. For example, the crawling controller 324 may be set such that it crawls a website regularly in a predetermined interval and/or it may start crawling when the website is modified. Also, users may set the extent of the crawling, i.e., users may set the website crawler 320 to crawl only within the original website from which the crawling is initiated, or allow crawling of web pages residing in external websites when web pages in the external websites are linked. In the latter case, it is desirable to limit the extent or depth of the crawling of the external websites. For example, in Figure 1, the system 100 may allow crawling of only website 1, allow crawling of web pages in secondary website 2 in addition to the originating website 1 only, or further allow crawling of tertiary websites 3 and 4.

[0045] Figure 6 describes the process of resolving script URLs by script execution in the context of website crawling in accordance with an embodiment of the present invention. The process will be described referring to the URL resolution system 300 shown in Figure 4. However, different systems, such as system 400 shown in Figure 5, may also be used.

[0046] The website crawler 320 crawls a website 310 (500). Crawling of website 310 may start anywhere in the website 310. During the crawling, the script code detector 322 checks script code embedded in each web page in the website 310 to determine if the web page uses script code to dynamically create one or more script URLs. When the script code detector 322 locates a web page with script code that dynamically creates one or more script URLs (502), the script URL resolution component 340 is activated. The script code detector 322 sends a notification to the script URL resolution component 340 to this end.

[0047]

In the script URL resolution component 340 the web page loading controller 342 instructs the web page examiner 360 to load the web page with the script code (504). The script code execution controller 344 then instructs the web page examiner 360 to execute the specific interface methods or functions that dynamically execute the script code and create one or more script URLs (506). The script code execution controller 344 may instruct the web page examiner 360 to execute the entire script code or only the relevant portions of the script code. Script URLs are thus resolved by the script code execution. The script code execution controller 344 receives the

resolved script URLs from the web page examiner 360, and sends the received script URLs back to the crawling controller 324 (508).

[0048] The website crawler 320 continues the crawling (510). It may continue crawling on web pages identified by the resolved script URLs. The website crawler 320 may crawl those web pages immediately when the resolved script URLs are returned, or put them in a queue for crawling at a later time. The website crawler 320 may crawl multiple web pages in parallel.

[0049] The process of Figure 6 represents the case where the links of the script URLs are extracted successfully. However, there may be situations where errors are encountered while executing the script code. Figure 7 depicts the process that occurs when the website crawler 320 encounters errors while executing the script code .

[0050] The steps of crawling a website (500) to executing script code (506) are similar to those shown in Figure 6. When the execution of the script code is successful, at least one script URL is resolved and obtained (520). The resolved script URL is reported back to the website crawler 320. In the website crawler 320, the crawling controller 324 controls the website crawler 320 to continue crawling the web page identified by the resolved script URL (510). The crawling is continued on the website containing the identified web page (524) immediately, or in parallel with the crawling of other web pages. Alternatively, the website containing the identified web page may be queued for crawling later in the scan or crawling process.

[0051] When the execution of the script code is unsuccessful, a failure result is output by the script URL resolution component 340 (530). The failure result is also returned to the website crawler 320. In the website crawler 320, the error result is logged (532), and the crawling of the current website is continued (534).

[0052] The process is repeated until crawling of the original website is completed.

[0053] The failure results logged at step 532 may be presented to users during and/or after the scanning.

[0054] Referring now to Figure 8, a URL resolution system 800 in accordance with another embodiment of the invention is described. The URL resolution system 800

comprises a website crawler 820 and an advanced web page examiner 860.

[0055] The website crawler 820 has a script URL gatherer 822 for gathering script URLs from the advanced web page examiner 860. The advanced web page examiner 860 has a web page loader 862 for loading web pages, and a script code examiner 864 for executing script code in the loaded web pages. The advanced web page examiner 860 may be a part of the URL resolution system 800 or a component external to the system 800.

[0056] In operation, the website crawler 820 crawls a website 810. For each URL found on each of those web pages, the script URL gatherer 822 calls a function on the advanced web page examiner 860. It also calls the function for the URL of each web page on which the website crawler 820 crawls. The function takes the received URL as an input parameter and activates the web page loader 862 to load the contents of a web page identified by the received URL.

[0057] Then the function activates the script code examiner 864 to examine the loaded web page to obtain any script URLs created by script code in the web page. For example, during the examination of the loaded web page, the script code examiner 864 executes script code found in the loaded web page to obtain script URLs if any. The script code examiner 864 may execute all script code in the loaded web page or only script code that is used to create one or more script URLs. Also, the script code examiner 864 may execute the entire script code or only relevant portions of script code.

[0058] The function returns a collection of zero or more resolved script URLs as an output parameter to the script URL gatherer 822. The website crawler 820 may crawl web pages identified by the resolved script URLs. The crawling of those web pages may be carried out immediately or later. The website crawler 820 may crawl those pages in parallel with other web pages.

[0059] The website crawler 820 may have a crawling controller similar to crawling controller 324 shown in Figure 4. Also, the URL resolution system 800 may have or use a presentation unit similar to Figure 4 or 5.

[0060] Figure 9 shows a modification of the URL resolution system 800 shown in Figure

8. In the modified URL resolution system 900, the website crawler 920 has a script code detector 924. Similarly to the script code detector 322 shown in Figure 4, the script code detector 924 checks if a web page contains script code that generates one or more script URLs. By using the script code detector 924, the website crawler 920 passes to the advanced web page examiner 860 only URLs of web pages that contain script code that generates one or more script URLs.

[0061] The advanced web page examiner 860 may be a part of the URL resolution system 900 or a component external to the system 900.

[0062] In the embodiments shown in Figures 4-9, the relevant parts of script code are explicitly executed to obtain script URLs. However, as described referring to Figure 3, script URLs may be obtained by examining script code, without explicit execution of the script code.

[0063] In the above embodiments, the elements of the URL resolution system are described separately, however, two or more elements may be provided as a single element, or one or more elements may be shared with other components in a computer system in which the URL resolution system is installed. For example, in the embodiment shown in Figure 2, the website crawler 120 and script URL resolution component 140 are shown as separate components. However, as shown in Figure 10, a URL resolution system 1000 may have a script URL resolution component 1040 as a part of website crawler 1020. A web page examiner 1060 may be a part of the URL resolution system 1000, or a separate component external to the system 1000. Furthermore, as shown in Figure 11, a URL resolution system 1100 may have a script URL resolution component 1140 and a web page examiner 1160 as components of website crawler 1120. Similar modifications may be made to the embodiments shown in Figures 4 and 5.

[0064] The URL resolution system of the present invention may be implemented by any hardware, software or a combination of hardware and software having the above described functions. The software code, either in its entirety or a part thereof, may be stored in computer readable memory. Further, a computer data signal representing the software code which may be embedded in a carrier wave may be transmitted via a communication network. Such a computer readable memory and a computer data

signal are also within the scope of the present invention, as well as the hardware, software and the combination thereof.

[0065] While particular embodiments of the present invention have been shown and described, changes and modifications may be made to such embodiments without departing from the true scope of the invention.